Technical Report: Environmental Variable Forecasting with Deep Learning

1. Problem Understanding & Dataset Analysis

Forecasting Objective

The primary objective of this project is to develop an accurate forecasting model for multiple environmental variables simultaneously, including average temperature, radiation, rainfall amount, wind speed, and wind direction. The model aims to predict these variables based on historical patterns and relationships between meteorological factors.

Dataset Analysis

The dataset consists of time series data split into training and testing sets, with daily observations of environmental variables over multiple years. Key characteristics include:

- Temporal data structured with year, month, and day columns (later converted to datetime)
- Multiple target variables: Avg_Temperature, Radiation, Rain_Amount, Wind_Speed, and Wind_Direction
- Potential data quality issues including missing values and out-of-range measurements
- Seasonal patterns in weather variables that need special consideration

Key Findings from Data Analysis

- The analysis revealed issues with date formatting that required correction, particularly with leap year handling (February 29)
- Missing values were present in both training and test datasets
- Some measurements contained physically impossible values (e.g., negative rainfall or wind speed)
- Temperature values potentially in different units (Kelvin vs. Celsius) required standardization
- Strong seasonal patterns were detected in the weather variables, suggesting the need for seasonal features

Preprocessing Justification

Several preprocessing steps were implemented to ensure data quality:

1. **Date handling**: Converting separate year, month, day columns into a proper datetime index while addressing leap year issues

2. **Missing value imputation**: Forward fill followed by backward fill approach, appropriate for time series data where values tend to be similar to adjacent timepoints
3. **Temperature standardization**: Converting values potentially in Kelvin to Celsius when above a threshold (100°C)
4. **Physical constraints enforcement**:
   - Negative rainfall and evapotranspiration values set to zero
   - Negative wind speeds converted to absolute values
   - Wind direction values constrained to 0-360° range
5. **Outlier handling**: Using IQR method with a 3× multiplier to identify and cap extreme values rather than removing them, preserving the time series continuity

2. Feature Engineering & Data Preparation

Feature Creation Techniques

An extensive set of features was engineered to capture various temporal patterns and variable interactions:

1. **Temporal features**:
   - Basic calendar features: month, day, day of year
   - Seasonal encoding with both categorical (Winter, Spring, Summer, Fall) and cyclical representation
   - Sine and cosine transformations for cyclical variables (month, day, day of year) to capture periodicity
2. **Lag features**:
   - Multiple time lags (1, 3, 7, 14 days) for all target variables to capture short and medium-term dependencies
   - These features provide the model with historical context essential for forecasting
3. **Rolling statistics**:
   - Rolling means and standard deviations with multiple windows (3, 7, 14 days)
   - These features capture local trends and volatility patterns
4. **Interaction terms**:
   - Temperature-rainfall interaction to capture combined effects
   - Wind-rainfall interaction to model complex weather dynamics

Feature Selection

Feature selection was implemented through correlation analysis:

- For each target variable, the top 15 most correlated features were identified
- The union of these important features across all targets was used for modeling
- This approach ensured that each prediction had access to the most relevant predictors
- Additional sine/cosine temporal features were included regardless of correlation to maintain seasonal context

Data Transformations

Several transformations were applied to ensure optimal model performance:

1. **Normalization**:
   - RobustScaler for input features to handle outliers effectively
   - MinMaxScaler for target variables to constrain outputs to a reasonable range (0-1)
2. **Sequence creation**:
   - 90-day sequences were created using a sliding window approach
   - This sequence length was selected to capture both short-term patterns and seasonal effects
3. **Train-validation split**:
   - 80% training, 20% validation chronological split preserving time series integrity
   - This approach respects the temporal nature of the data without data leakage

3. Model Selection & Justification

Model Evaluation

While the code focuses on a deep learning approach, implicit consideration of model selection is evident:

1. **Selected model**: Bidirectional LSTM neural network
   - Well-suited for capturing complex temporal dependencies in multivariate time series
   - Capable of modeling non-linear relationships between variables
2. **Architecture justification**:
   - Bidirectional LSTM layers to capture both past and future context within sequences
   - Multiple stacked layers (two bidirectional LSTM layers followed by a unidirectional layer) for hierarchical feature extraction
   - Dropout layers (0.2-0.3) to prevent overfitting
   - Dense layers for final prediction with sufficient capacity

Hyperparameter Optimization

The model incorporates several optimized hyperparameters:

1. **Sequence length**: 90 days, selected to capture seasonal patterns
2. **Layer architecture**: 128 → 96 → 64 → 48 units, gradually reducing complexity
3. **Learning rate**: 0.001 with adaptive reduction
4. **Loss function**: Huber loss for robustness against outliers
5. **Batch size**: 32 for stable gradient updates

The optimization strategy includes:

- ReduceLROnPlateau callback to adaptively decrease learning rate when validation performance plateaus
- EarlyStopping to prevent overfitting with a patience of 10 epochs and best weight restoration

Validation Approach

A time-based validation strategy was implemented:

- Chronological train-test split (80/20) preserving temporal order
- Walk-forward validation for test predictions, where the model incrementally updates its input sequence with each prediction
- This approach mimics real-world forecasting scenarios where only past data is available

4. Performance Evaluation & Error Analysis

Evaluation Metrics

Multiple complementary metrics were used to assess model performance:

1. **SMAPE (Symmetric Mean Absolute Percentage Error)**:
   - Scale-independent metric suitable for comparing accuracy across different variables
   - Less affected by outliers and avoids division by zero issues
2. **RMSE (Root Mean Squared Error)**:
   - Penalizes larger errors more heavily
   - Useful for understanding the magnitude of prediction errors
3. **MAE (Mean Absolute Error)**:
   - Provides average error magnitude in the original units

- Easier to interpret directly in the context of each variable

These metrics were calculated separately for each target variable, acknowledging their different scales and importance.

Model Performance Analysis

The model's performance was visualized through:

- Loss curves comparing training and validation loss across epochs
- Direct comparison plots of actual vs. predicted values for each target variable
- These visualizations help identify potential overfitting, underfitting, or systematic bias

Residual Analysis

While not explicitly coded, the visualization of actual vs. predicted values enables visual inspection for:

- Systematic prediction errors
- Areas where the model struggles (particular weather patterns or seasons)
- Variable-specific performance issues

Model Limitations

Based on the implementation, several limitations can be identified:

1. **Data constraints**:
   - Reliance on historical patterns may limit performance during extreme or unusual weather events
   - Potential gaps in capturing complex interactions between variables
2. **Prediction stability**:
   - Walk-forward validation may lead to error accumulation over longer forecast horizons
   - Post-processing with rolling means was required to reduce unrealistic fluctuations
3. **Computational complexity**:
   - The bidirectional LSTM architecture, while powerful, requires significant training time
   - The model's complexity might be excessive for some simpler seasonal patterns

5. Interpretability & Business Insights

Real-World Applications

The forecasting model has several practical applications:

1. **Agricultural planning**:
   - Crop management based on temperature and rainfall forecasts
   - Irrigation scheduling using predicted evapotranspiration and rainfall
2. **Energy sector**:
   - Renewable energy production forecasting (solar based on radiation, wind based on speed/direction)
   - Demand forecasting for heating/cooling based on temperature predictions
3. **Disaster preparedness**:
   - Early warning systems for heavy rainfall events
   - Wind hazard predictions for infrastructure management
4. **Water resource management**:
   - Reservoir operations based on rainfall and temperature forecasts
   - Drought monitoring and preparedness

Improvement Suggestions

1. **Model enhancements**:
   - Consider ensemble methods combining LSTM with traditional statistical models
   - Experiment with attention mechanisms to better capture long-term dependencies
   - Implement uncertainty estimation through prediction intervals
2. **Feature engineering improvements**:
   - Incorporate external data sources such as large-scale climate indices (ENSO, NAO)
   - Add spatial features if the location information is available
   - Develop more sophisticated interaction terms based on meteorological principles
3. **Deployment strategy**:
   - Implement automated retraining pipeline with new data incorporation
   - Develop monitoring system for detecting prediction drift
   - Create visualization dashboards for non-technical stakeholders
   - Consider model compression techniques for faster inference
4. **Domain-specific customizations**:
   - Calibrate model outputs with physics-based constraints
   - Optimize predictions for specific use cases (e.g., extreme event prediction vs. average conditions)
   - Develop specialized models for critical variables or seasons